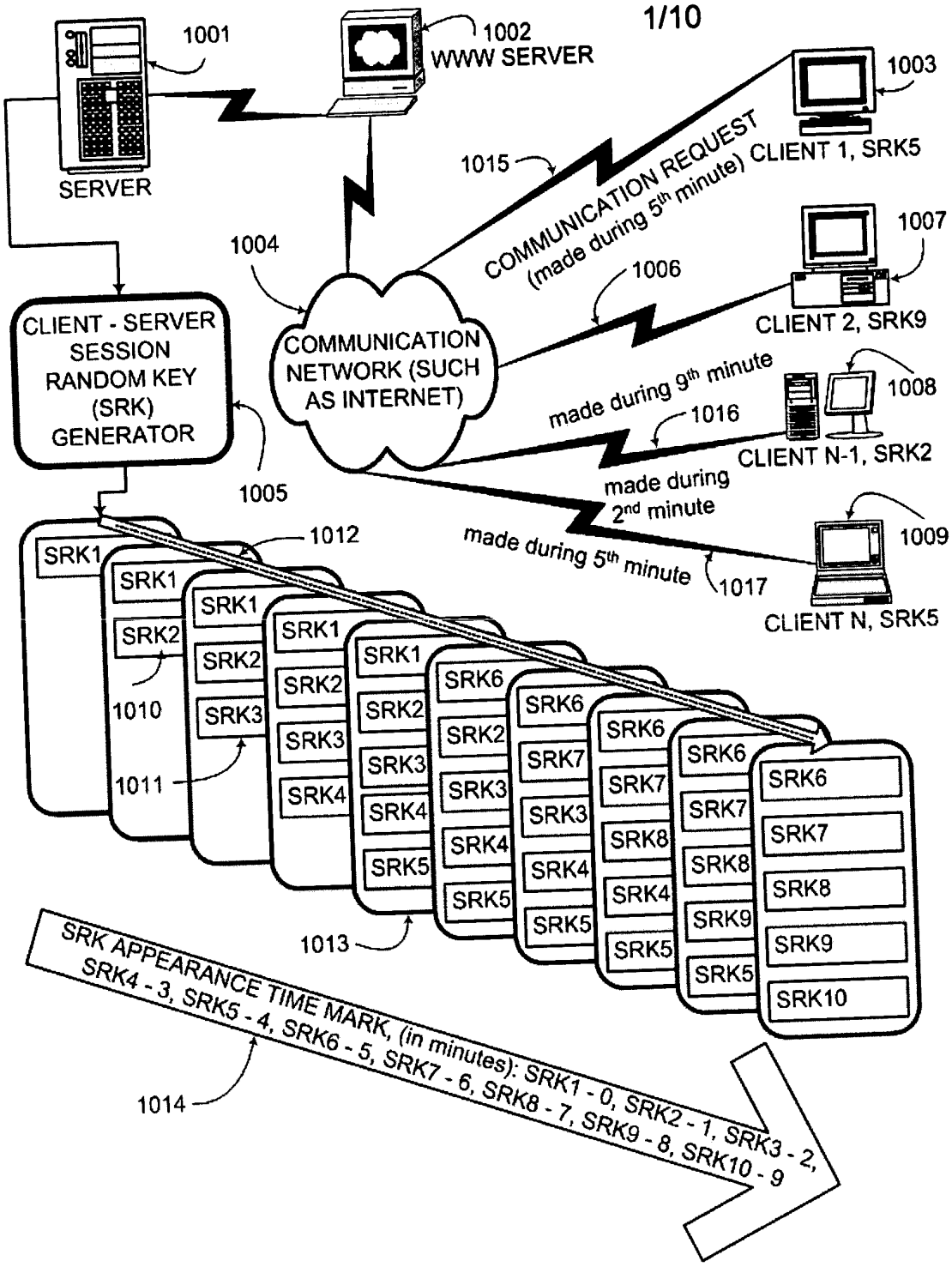
1001
SERVER
1002
WWW SERVER
1003
CLIENT 1, SRK5
1015
COMMUNICATION REQUEST
(made during 5th minute)
1004
1006
1007
CLIENT 2, SRK9
CLIENT - SERVER SESSION RANDOM KEY (SRK) GENERATOR
COMMUNICATION NETWORK (SUCH AS INTERNET)
made during 9th minute
1016
1008
CLIENT N-1, SRK2
made during 2nd minute
1005
1012
made during 5th minute
1017
1009
CLIENT N, SRK5
SRK1
SRK2
SRK3
SRK4
SRK5
SRK6
SRK7
SRK8
SRK9
SRK10
1010
1011
1013
SRK APPEARANCE TIME MARK, (in minutes): SRK1 - 0, SRK2 - 1, SRK3 - 2, SRK4 - 3, SRK5 - 4, SRK6 - 5, SRK7 - 6, SRK8 - 7, SRK9 - 8, SRK10 - 9
1014

FIG. 1

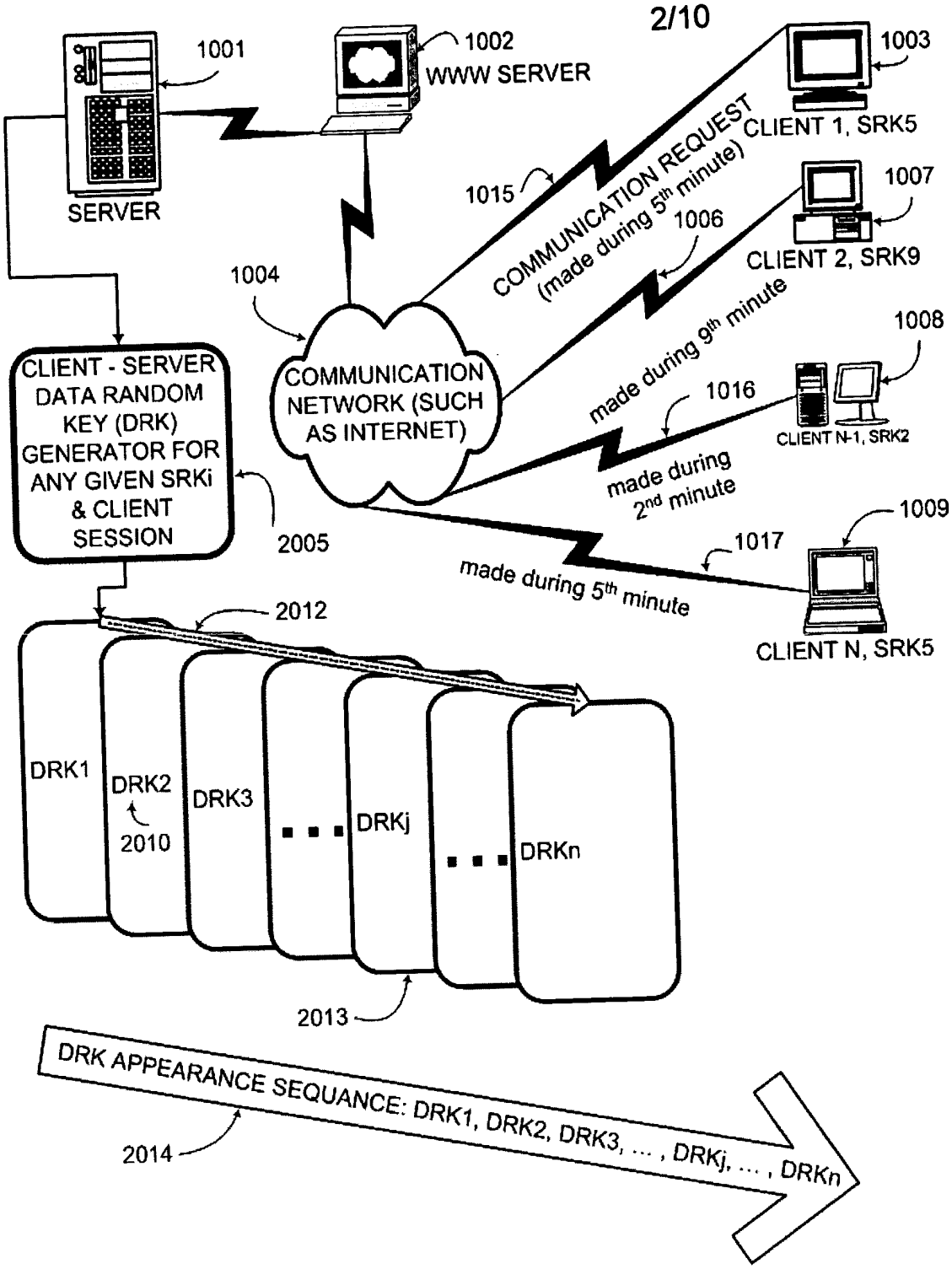
1001
SERVER
1002
WWW SERVER
1003
CLIENT 1, SRK5
1004
COMMUNICATION NETWORK (SUCH AS INTERNET)
1015
COMMUNICATION REQUEST (made during 5th minute)
1006
1007
CLIENT 2, SRK9
made during 9th minute
1008
CLIENT N-1, SRK2
1016
made during 2nd minute
1009
1017
made during 5th minute
CLIENT N, SRK5
CLIENT - SERVER DATA RANDOM KEY (DRK) GENERATOR FOR ANY GIVEN SRKi & CLIENT SESSION
2005
2012
DRK1
DRK2
2010
DRK3
DRKj
DRKn
2013
DRK APPEARANCE SEQUANCE: DRK1, DRK2, DRK3, ... , DRKj, ... , DRKn
2014

FIG. 2

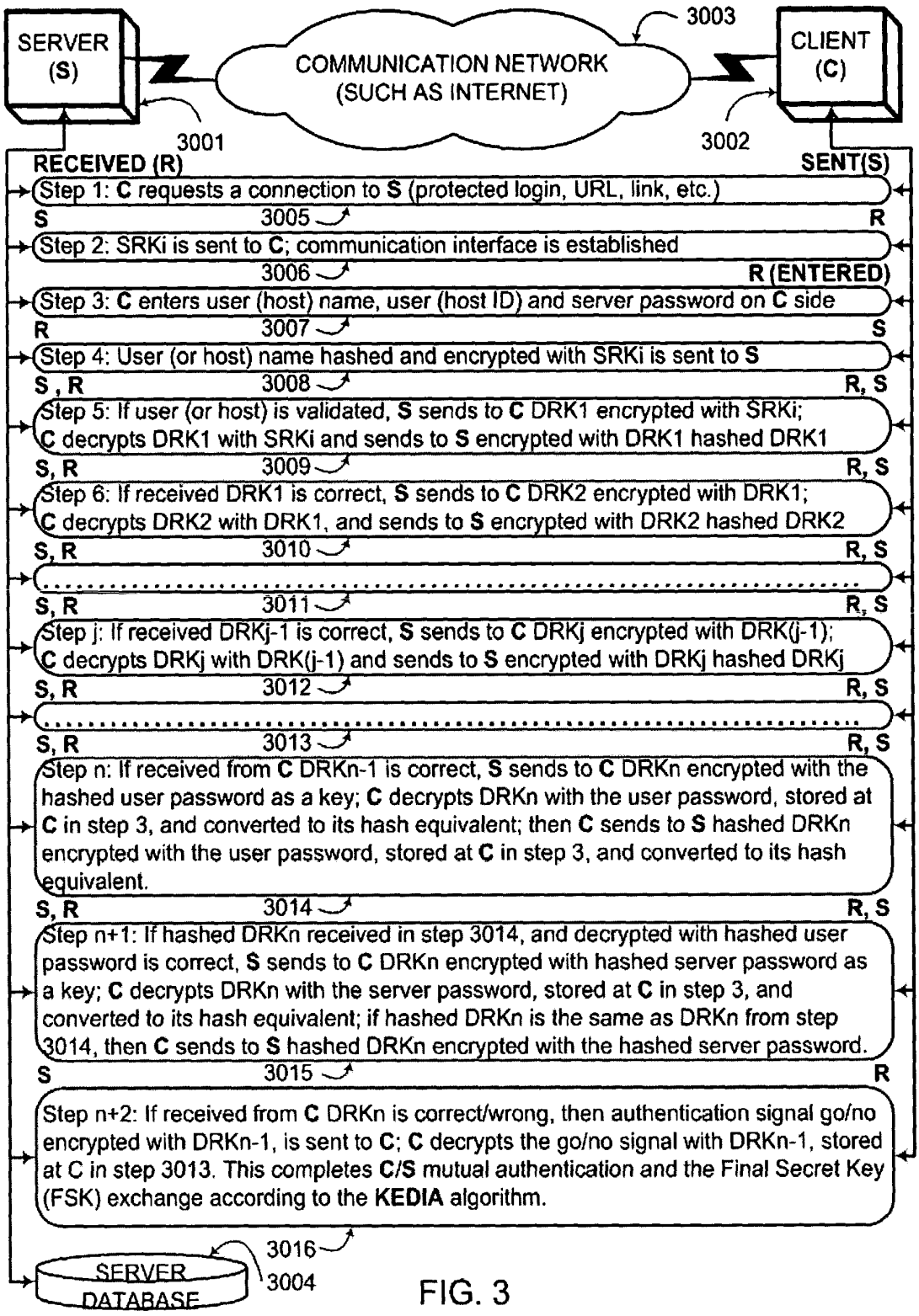
SERVER (S)
COMMUNICATION NETWORK (SUCH AS INTERNET)
3003
CLIENT (C)
3001
3002
RECEIVED (R)
SENT(S)
Step 1: C requests a connection to S (protected login, URL, link, etc.)
S
3005
R
Step 2: SRKi is sent to C; communication interface is established
3006
R (ENTERED)
Step 3: C enters user (host) name, user (host ID) and server password on C side
R
3007
S
Step 4: User (or host) name hashed and encrypted with SRKi is sent to S
S , R
3008
R, S
Step 5: If user (or host) is validated, S sends to C DRK1 encrypted with SRKi;
C decrypts DRK1 with SRKi and sends to S encrypted with DRK1 hashed DRK1
S, R
3009
R, S
Step 6: If received DRK1 is correct, S sends to C DRK2 encrypted with DRK1;
C decrypts DRK2 with DRK1, and sends to S encrypted with DRK2 hashed DRK2
S, R
3010
R, S
S, R
3011
R, S
Step j: If received DRKj-1 is correct, S sends to C DRKj encrypted with DRK(j-1);
C decrypts DRKj with DRK(j-1) and sends to S encrypted with DRKj hashed DRKj
S, R
3012
R, S
S, R
3013
R, S
Step n: If received from C DRKn-1 is correct, S sends to C DRKn encrypted with the hashed user password as a key; C decrypts DRKn with the user password, stored at C in step 3, and converted to its hash equivalent; then C sends to S hashed DRKn encrypted with the user password, stored at C in step 3, and converted to its hash equivalent.
S, R
3014
R, S
Step n+1: If hashed DRKn received in step 3014, and decrypted with hashed user password is correct, S sends to C DRKn encrypted with hashed server password as a key; C decrypts DRKn with the server password, stored at C in step 3, and converted to its hash equivalent; if hashed DRKn is the same as DRKn from step 3014, then C sends to S hashed DRKn encrypted with the hashed server password.
S
3015
R
Step n+2: If received from C DRKn is correct/wrong, then authentication signal go/no encrypted with DRKn-1, is sent to C; C decrypts the go/no signal with DRKn-1, stored at C in step 3013. This completes C/S mutual authentication and the Final Secret Key (FSK) exchange according to the KEDIA algorithm.
3016
SERVER DATABASE
3004

FIG. 3

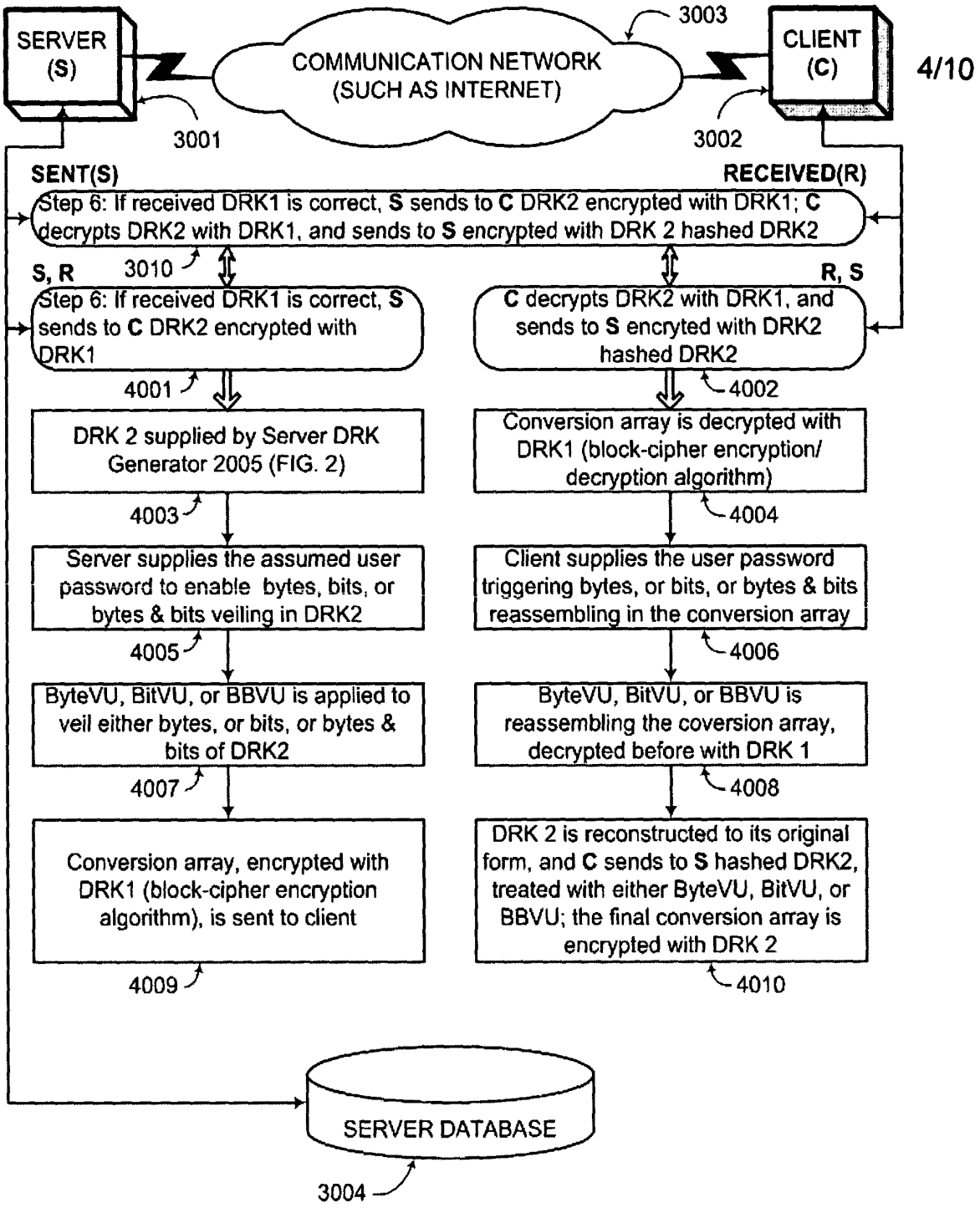
SERVER (S)
COMMUNICATION NETWORK (SUCH AS INTERNET)
3003
CLIENT (C)
3001
3002
SENT(S)
RECEIVED(R)
Step 6: If received DRK1 is correct, S sends to C DRK2 encrypted with DRK1; C decrypts DRK2 with DRK1, and sends to S encrypted with DRK 2 hashed DRK2
S, R
3010
R, S
Step 6: If received DRK1 is correct, S sends to C DRK2 encrypted with DRK1
C decrypts DRK2 with DRK1, and sends to S encryted with DRK2 hashed DRK2
4001
4002
DRK 2 supplied by Server DRK Generator 2005 (FIG. 2)
Conversion array is decrypted with DRK1 (block-cipher encryption/ decryption algorithm)
4003
4004
Server supplies the assumed user password to enable bytes, bits, or bytes & bits veiling in DRK2
Client supplies the user password triggering bytes, or bits, or bytes & bits reassembling in the conversion array
4005
4006
ByteVU, BitVU, or BBVU is applied to veil either bytes, or bits, or bytes & bits of DRK2
ByteVU, BitVU, or BBVU is reassembling the coversion array, decrypted before with DRK 1
4007
4008
Conversion array, encrypted with DRK1 (block-cipher encryption algorithm), is sent to client
DRK 2 is reconstructed to its original form, and C sends to S hashed DRK2, treated with either ByteVU, BitVU, or BBVU; the final conversion array is encrypted with DRK 2
4009
4010
SERVER DATABASE
3004

FIG. 4

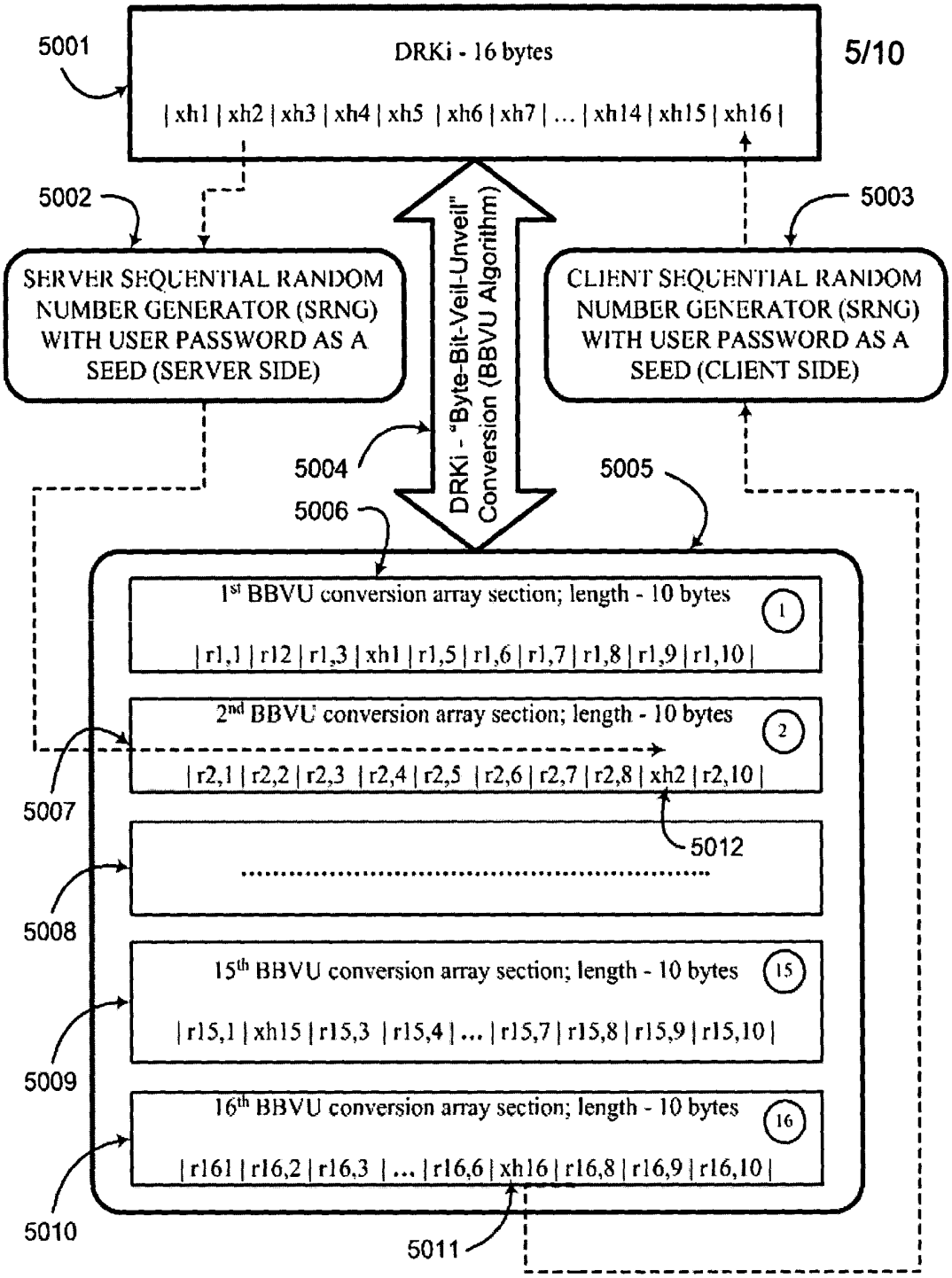
5001
DRKi - 16 bytes
| xh1 | xh2 | xh3 | xh4 | xh5 | xh6 | xh7 | ... | xh14 | xh15 | xh16 |
5002
5003
SERVER SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER PASSWORD AS A SEED (SERVER SIDE)
CLIENT SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER PASSWORD AS A SEED (CLIENT SIDE)
DRKi - "Byte-Bit-Veil-Unveil" Conversion (BBVU Algorithm)
5004
5005
5006
1st BBVU conversion array section; length - 10 bytes
1
| r1,1 | r12 | r1,3 | xh1 | r1,5 | r1,6 | r1,7 | r1,8 | r1,9 | r1,10 |
2nd BBVU conversion array section; length - 10 bytes
2
| r2,1 | r2,2 | r2,3 | r2,4 | r2,5 | r2,6 | r2,7 | r2,8 | xh2 | r2,10 |
5007
5012
.....
5008
15th BBVU conversion array section; length - 10 bytes
15
| r15,1 | xh15 | r15,3 | r15,4 | ... | r15,7 | r15,8 | r15,9 | r15,10 |
5009
16th BBVU conversion array section; length - 10 bytes
16
| r161 | r16,2 | r16,3 | ... | r16,6 | xh16 | r16,8 | r16,9 | r16,10 |
5010
5011

FIG. 5

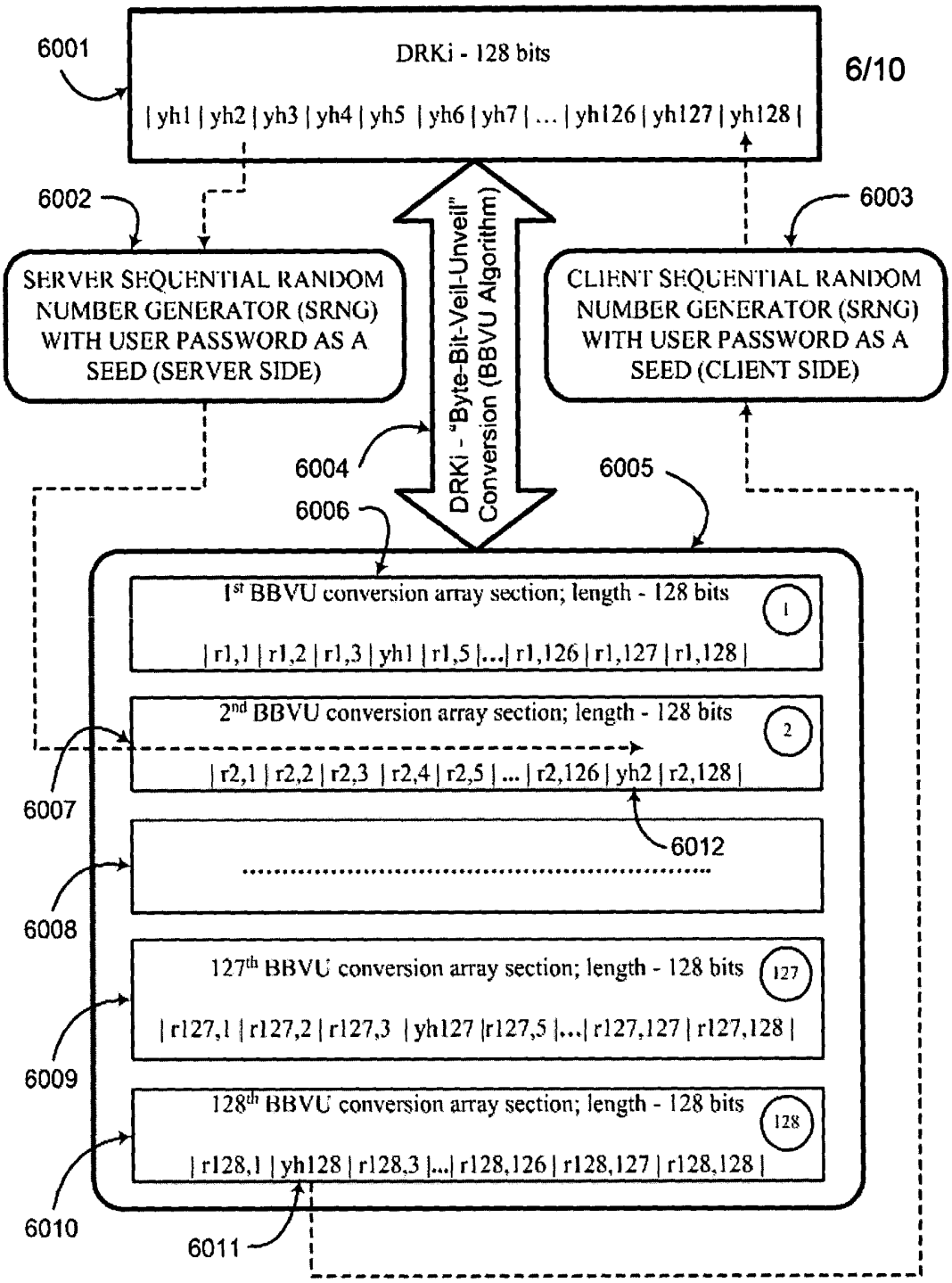
6001
DRKi - 128 bits
| yh1 | yh2 | yh3 | yh4 | yh5 | yh6 | yh7 | ... | yh126 | yh127 | yh128 |
6002
6003
SERVER SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER PASSWORD AS A SEED (SERVER SIDE)
CLIENT SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER PASSWORD AS A SEED (CLIENT SIDE)
DRKi - "Byte-Bit-Veil-Unveil" Conversion (BBVU Algorithm)
6004
6005
6006
1st BBVU conversion array section; length - 128 bits
1
| r1,1 | r1,2 | r1,3 | yh1 | r1,5 |...| r1,126 | r1,127 | r1,128 |
2nd BBVU conversion array section; length - 128 bits
2
| r2,1 | r2,2 | r2,3 | r2,4 | r2,5 | ... | r2,126 | yh2 | r2,128 |
6007
6012
6008
127th BBVU conversion array section; length - 128 bits
127
| r127,1 | r127,2 | r127,3 | yh127 |r127,5 |...| r127,127 | r127,128 |
6009
128th BBVU conversion array section; length - 128 bits
128
| r128,1 | yh128 | r128,3 |...| r128,126 | r128,127 | r128,128 |
6010
6011

FIG. 6

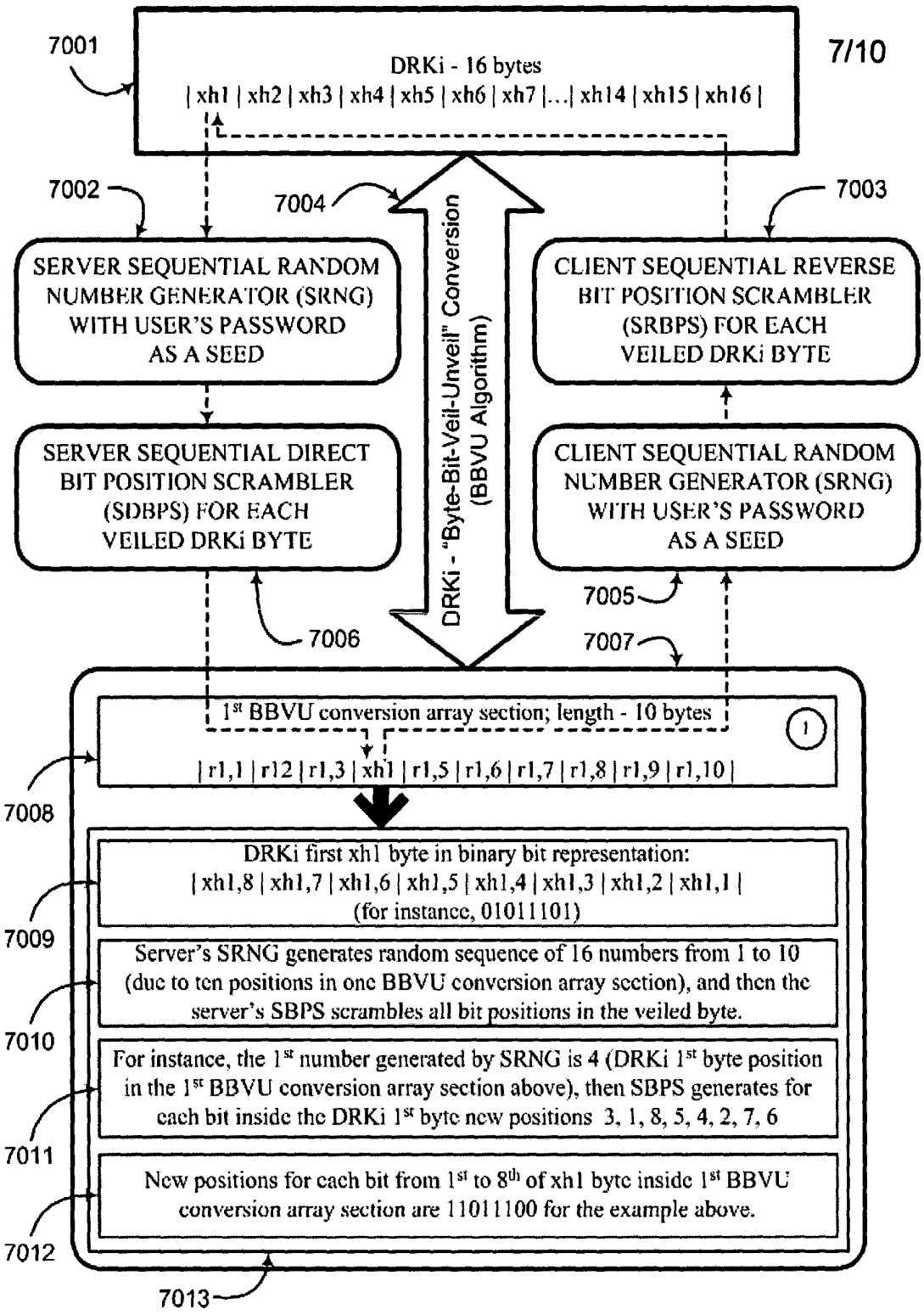
7001
DRKi - 16 bytes
| xh1 | xh2 | xh3 | xh4 | xh5 | xh6 | xh7 |...| xh14 | xh15 | xh16 |
7/10
7002
7004
7003
SERVER SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER'S PASSWORD AS A SEED
DRKi - "Byte-Bit-Veil-Unveil" Conversion (BBVU Algorithm)
CLIENT SEQUENTIAL REVERSE BIT POSITION SCRAMBLER (SRBPS) FOR EACH VEILED DRKi BYTE
SERVER SEQUENTIAL DIRECT BIT POSITION SCRAMBLER (SDBPS) FOR EACH VEILED DRKi BYTE
CLIENT SEQUENTIAL RANDOM NUMBER GENERATOR (SRNG) WITH USER'S PASSWORD AS A SEED
7005
7006
7007
1st BBVU conversion array section; length - 10 bytes
1
| r1,1 | r12 | r1,3 | xh1 | r1,5 | r1,6 | r1,7 | r1,8 | r1,9 | r1,10 |
7008
DRKi first xh1 byte in binary bit representation:
| xh1,8 | xh1,7 | xh1,6 | xh1,5 | xh1,4 | xh1,3 | xh1,2 | xh1,1 |
(for instance, 01011101)
7009
Server's SRNG generates random sequence of 16 numbers from 1 to 10 (due to ten positions in one BBVU conversion array section), and then the server's SBPS scrambles all bit positions in the veiled byte.
7010
For instance, the 1st number generated by SRNG is 4 (DRKi 1st byte position in the 1st BBVU conversion array section above), then SBPS generates for each bit inside the DRKi 1st byte new positions 3, 1, 8, 5, 4, 2, 7, 6
7011
New positions for each bit from 1st to 8th of xh1 byte inside 1st BBVU conversion array section are 11011100 for the example above.
7012
7013

FIG. 7

MESSAGE ENCRYPT/DECRYPT ITERATIVE AUTHENTICATION (MEDIA) PROTOCOL
(Client - Server Architecture; Password Based Mutual Authentication; Part A - SERVER SIDE)

8000 ↓ #	8001 ↓ SERVER	8002 ↓ SENT	8003 ↓ RECEIVED	8004 ↓ E-KEY	8005 ↓ D-KEY	8006 ↓ ByteVU **CA**, SECTIONS, BYTES	8007 ↓ SRNG SEED	8008 ↓ C-LINE
1	www **s**		**c** connection request			no		to SERVER
2	servlets	SRK (compiled)				no		to CLIENT
3	dB/servlets		h-u-name		SRK	no		to SERVER
4	dB/servlets	DRK1		SRK		yes, 16, 25	h-u-password	to CLIENT
5	dB/servets		h-DRK1		DRK1	yes, 16, 25	h-u-password	to SERVER
5A	dB/servlets	no		SRK		no		to CLIENT
6	dB/servlets	DRK2		DRK1		yes, 16, 25	h-u-password	to CLIENT
7	dB/servlets		h-DRK2		DRK2	yes, 16, 25	h-u-password	to SERVER
8	dB/servlets	DRK2		DRK2		yes, 16, 25	h-**s**-password	to CLIENT
9	dB/servlets		h-DRK2		DRK2	yes, 16, 25	h-**s**-password	to SERVER
10	dB/servlets	go/no		DRK1		yes, 16, 25	h-u-password	to CLIENT

8017→ Legends: SRNG - Sequential Random Number Generator,
SRK (DRK) - Session (Data) Random Symmetric Encryption Key, ByteVU - "Byte-Veil-Unveil" Algorithm,
E - ENCRYPTION, D - DECRYPTION, C - COMMUNICATION, u - user, h - hashed, p - protected
c - Client, **s** - Server, dB - Database, **CA** - ByteVU Conversion Array

FIG. 8A

Inventor: Len L. Miller
Title: Key Conversion Method For Communication Session Encryption And Authentication System
Application No. _____
Attorney Docket No. AIDT 1006-1

MESSAGE ENCRYPT/DECRYPT ITERATIVE AUTHENICATION (MEDIA) PROTOCOL
(Client - Server Architecture; Password Based Mutual Authentication; Part B - CLIENT SIDE)

8008 ↓ C-LINE	8009 ↓ SRNG SEED	8010 ↓ ByteVU **CA**, SECTIONS, BYTES	8011 ↓ D-KEY	8012 ↓ E-KEY	8013 ↓ ENTERED BY CLIENT	8014 ↓ RECEIVED	8015 ↓ SENT	8016 ↓ CLIENT	8000 ↓ #
to SERVER		no					p-link / URL	WAN/LAN	1
to CLIENT		no				GUI, SRK (compiled)		applet logic	2
to SERVER		no		SRK	u-name, **s**- & u-passwords		h-u-name	applet GUI	3
to CLIENT	h-u-password	yes, 16, 25	SRK			DRK1		applet logic	4
to SERVER	h-u-password	yes, 16, 25		DRK1			h-DRK1		5
to CLIENT		no	SRK			error message		applet GUI	5A
to CLIENT	h-u-password	yes, 16, 25	DRK1			DRK2		applet logic	6
to SERVER	h-u-password	yes, 16, 25		DRK2			h-DRK2		7
to CLIENT	h-**s**-password	yes, 16, 25	DRK2			h-DRK2		applet logic	8
to SERVER	h-**s**-password	yes, 16, 25		DRK2			h-DRK2		9
to CLIENT	h-u-password	yes, 16, 25	DRK1			go/no		applet GUI	10

8017→ Legends: SRNG - Sequential Random Number Generator,
SRK (DRK) - Session (Data) Random Symmetric Encryption Key, ByteVU - "Byte-Veil-Unveil" Algorithm,
E - ENCRYPTION, D - DECRYPTION, C - COMMUNICATION, u - user, h - hashed, p - protected
c - Client, **s** - Server, **CA** - ByteVU Conversion Array

FIG. 8B

Inventor: [illegible]
Title: Key Conversion Method For Communication Session Encryption And Authentication System
Application No. _____
Attorney Docket No. AIDT 1006-1

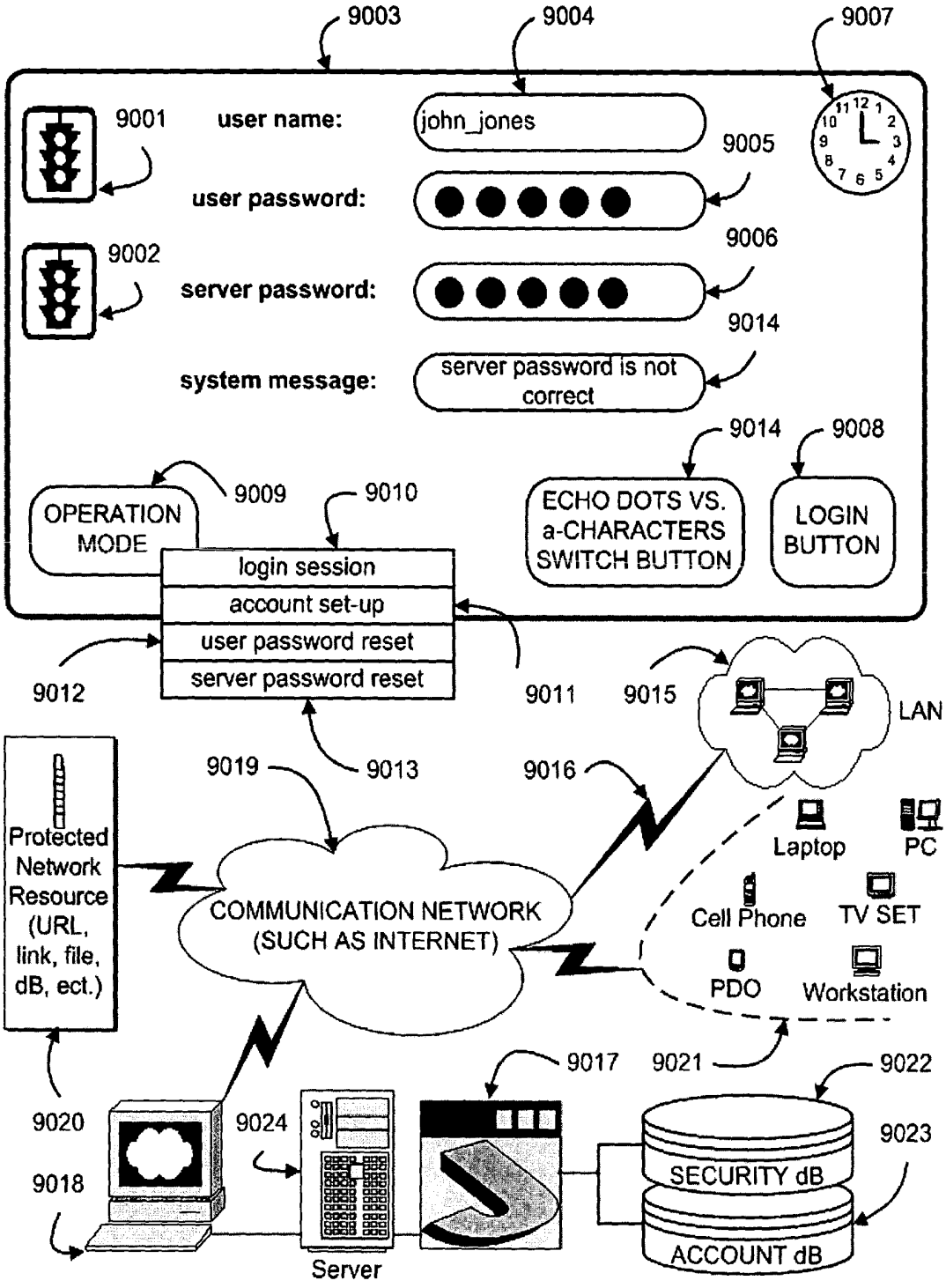
9003
9004
9007
9001
user name:
john_jones
9005
user password:
9002
server password:
9006
9014
system message:
server password is not correct
9014
9008
OPERATION MODE
9009
9010
ECHO DOTS VS. a-CHARACTERS SWITCH BUTTON
LOGIN BUTTON
login session
account set-up
user password reset
server password reset
9012
9011
9015
LAN
9019
9013
9016
Protected Network Resource (URL, link, file, dB, ect.)
COMMUNICATION NETWORK (SUCH AS INTERNET)
Laptop
PC
Cell Phone
TV SET
PDO
Workstation
9017
9021
9022
9020
9024
9023
9018
SECURITY dB
ACCOUNT dB
Server

FIG. 9